

NAKI-AN73
MASATO SUZUKI et al.

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

紙添付の書類に記載されている事項は下記の出願書類に記載されて
事項と同一であることを証明する。

is to certify that the annexed is a true copy of the following application as filed
his Office.

願年月日
of Application:

1993年 5月31日

願番号
Application Number:

平成 5年特許願第129529号

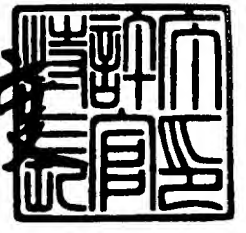
願人
Applicant(s):

松下電器産業株式会社

1994年 3月11日

特許庁長官
Commissioner.
Patent Office

麻生 渡



出証番号 出証特平06-3008647

【書類名】 特許願

【整理番号】 2030452161

【提出日】 平成 5年 5月31日

【あて先】 特許庁長官 麻 生 渡 殿

【国際特許分類】 G06F 9/30

【発明の名称】 データ処理装置およびデータ処理方法

【請求項の数】 6

【発明者】

【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

【氏名】 神山 祐史

【発明者】

【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

【氏名】 鈴木 正人

【発明者】

【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

【氏名】 宮地 信哉

【特許出願人】

【識別番号】 000005821

【氏名又は名称】 松下電器産業株式会社

【代表者】 森下 洋一

【代理人】

【識別番号】 100090446

【弁理士】

【氏名又は名称】 中島 司朗

【手数料の表示】

【納付方法】 予納

【予納台帳番号】 014823

【納付金額】 14,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9003742

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ処理装置およびデータ処理方法

【特許請求の範囲】

【請求項1】 即値データを指定した命令に従って所定のデータ演算を行うデータ処理装置であって、

整数Nビット幅を有する第1レジスタと、

Nビット幅を有する第2レジスタと、

Nより小さいM（整数）ビットデータの最上位ビットを上位側へ複写することによって前記Mビットデータのデータ幅をNビットに拡張する符号拡張器と、

Mビットデータの上位側にゼロ値を発生させることによって前記Mビットデータのデータ幅をNビットに拡張するゼロ拡張器と、

与えられた命令に従って演算処理を行う演算器と、

命令を解読し、前記命令がMビットの即値データを含み、かつ前記第1レジスタに対する格納処理を有する場合、前記即値データを前記ゼロ拡張器に出力し、前記命令がMビットの即値データを含み、かつ前記第2レジスタに対する格納処理を有する場合、前記即値データを前記符号拡張器に出力するように前記第1レジスタ、前記第2レジスタ及び前記演算器を制御する命令制御手段とを備えた、データ処理装置。

【請求項2】 前記第1レジスタは、メモリのアドレスデータを格納するアドレスレジスタであり、

前記第2レジスタは、前記命令のデータオペランドを格納するデータレジスタである、請求項1記載のデータ処理装置。

【請求項3】 前記命令制御手段は、2のべき乗の即値データを解読する、請求項1記載のデータ処理装置。

【請求項4】 整数Nビット幅の第1及び第2レジスタに対してアクセスするNより小さい整数Mビット幅の即値データを指定した命令に従って所定の演算を行うデータ処理方法であって、

命令を解読し、命令が指定するアクセス先レジスタを判別する命令解読ステップと、

前記命令が前記第1レジスタに対する格納処理を有する場合、Mビット幅の前記即値データをゼロ拡張することによってNビット幅のデータに拡張し、前記命令が前記第2レジスタに対する格納処理を有する場合、Mビット幅の前記即値データを符号拡張することによってNビット幅のデータに拡張するデータ拡張ステップと、

Nビット幅に拡張された前記即値データを用い、前記命令に従って所定の演算処理を行う演算ステップとを備えた、データ処理方法。

【請求項5】 前記第1レジスタは、メモリのアドレスデータを格納するアドレスレジスタであり、

前記第2レジスタは、前記命令のデータオペランドを格納するデータレジスタである、請求項5記載のデータ処理方法。

【請求項6】 前記即値データのビット幅Mは、2のべき乗で指定される、請求項4記載のデータ処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は、プログラムの命令に従って処理を実行するデータ処理装置及びデータ処理方法に関し、特に、メモリにアクセスするためのアドレス空間におけるアドレスを指定する即値データ（即値アドレス）の指定処理あるいは演算処理の適正化が可能なデータ処理装置およびデータ処理方法に関するものである。

【0002】

【従来の技術】

一般的に、マイクロコンピュータ等のデータ処理装置では、種々のデータ幅を有するデータの処理が可能のように構成されている。例えば、プログラムの命令によって直接的に数値を指定する即値データに対しては、符号拡張処理によってデータ幅の調整がなされた後、所定の演算処理が行われるように構成されている。

【0003】

以下、この即値データに対する符号拡張処理を伴うデータ処理装置の構成及び

動作について説明する。

即値データの符号拡張機能を有する従来のデータ処理装置の例が、”「マイクロコンピュータシリーズ14 68000マイクロコンピュータ」喜田祐三他、丸善1983年3月”に記載されている。図7は、この従来のデータ処理装置の構成を示すブロック図である。図7において、11は主にデータを格納するN（整数）ビット幅のデータレジスタ群、12は主にアドレスを格納するNビット幅のアドレスレジスタ群、13は16ビットデータの最上位ビットを符号拡張して32ビットデータとして出力する符号拡張器、14は命令を解読する命令解読部、15は命令解読部14での解読結果に基づいて演算を実行する演算器である。

【0004】

この従来のデータ処理装置の一般的な動作を以下に説明する。

まず命令解読部14は、外部から与えられた命令を解読する。そして、命令の内容によって次のような動作を行う。

（1）命令が「データレジスタ11またはアドレスレジスタ12間でデータを格納し、あるいは演算する」場合。

【0005】

演算器15に2つのNビット幅のデータが入力され、所定の演算がなされた後、指定されたレジスタにデータが格納される。

（2）命令が「Mビットの即値データをデータレジスタ11またはアドレスレジスタ12に格納し、あるいは演算する」場合。

Mビットの即値データが符号拡張器13でNビットに拡張されて演算器15の一方の入力となり、所定の演算が行われた後、指定されたレジスタへ出力される。

【0006】

ここで、符号拡張器13の動作を図5を用いて説明する。まず、図5（a）において、Mビットで表現される数値の最上位ビットが0の場合には、上位側へNビットまで0を複写してデータ幅を拡張する。また、図5（b）において、Mビットで表現される数値の最上位ビットが1の場合には、上位側へNビットまで1を複写してデータ幅を拡張する。

【0007】

さらに、この従来のデータ処理装置において、特に、即値データのビット幅が格納または演算する対象となるレジスタのビット幅よりも短い場合の即値データの処理動作について具体的に説明する。一例として、図8に示すプログラムの実行動作について説明する。図8に示すプログラムは、32ビットでアドレス指定される全空間の中でH' 8000（H' は16進数表記を示す）番地からH' 8100番地までH' 10番地刻みに格納されているメモリ上の16個のデータを順次加算し、その和をH' 10000000番地へ書き出す処理を示している。図9は、図8においてニモニク表記されたプログラムの処理フローを示している。さらに、図10は、メモリのアドレス空間を模式的に示している。以下、図7～図10を参照して説明する。なお、以下の説明では、即値データの符号拡張処理は、正規の拡張処理が行われる場合を示しており、実際の符号拡張処理における問題点については、後述される。

命令（1）：データレジスタD0をクリアする。

命令（2）：アドレスレジスタA0に32ビットの即値データH' 8000を設定する。

【0008】

この命令の表記は16ビットの即値データH' 8000を使用している。したがって、この即値データH' 8000は、符号拡張器13によって32ビットデータに符号拡張された後、アドレスレジスタA0に格納される。

命令（3）：アドレスレジスタA0で指定される番地の内容をメモリから読み出してデータレジスタD1に格納する。

命令（4）：データレジスタD1の内容をデータレジスタD0の内容に加算し、その結果をデータレジスタD0に格納する。

命令（5）：アドレスレジスタA0に即値H' 0010を加算し、アドレスレジスタA0に格納する。

【0009】

16ビット即値データH' 0010は、符号拡張器13によって32ビットデータH' 00000010に拡張される。その後、演算器15においてアド

レスレジスタA0に格納されているアドレスデータH' 00008000と拡張されたデータH' 00000010とが加算され、H' 00008010が算出される。この算出データはアドレスレジスタA0に格納される。

命令(6) : 上記の算出データと即値データH' 8100とを比較する。

【0010】

即値データH' 8100は符号拡張器13によって32ビットデータに符号拡張された後、演算器15に出力される。そして、演算器15において、アドレスレジスタA0から読み出されたアドレスデータH' 00008010と比較される。

命令(7) : 比較結果に基づいて、小さければラベルAで示す命令(3)へ戻り、等しいかまたは大きければ命令(8)へ進むように判断する。

【0011】

命令(3)から命令(7)で記述されるループ処理は、アドレスレジスタA0の値がH' 00008000から始めてH' 00000010ずつ更新されH' 00008100になるまで繰返されるために、16回の加算結果がデータレジスタD0に格納された後、命令(8)へ進む。

命令(8) : データレジスタD0の内容をメモリのH' 10000000番地へ格納する。

【0012】

【発明が解決しようとする課題】

しかしながら、上記の従来のデータ処理装置においては、アドレスレジスタ12に対してアクセスされる即値データについて行われた符号拡張によって、指定した即値データの内容と異なるデータとなる場合が生じる。このような状態を図8に示す例で説明すると、符号拡張される即値データは、命令(2)におけるH' 8000と、命令(5)におけるH' 0010と、命令(6)におけるH' 8100である。

【0013】

まず、命令(2)における即値データH' 8000は、符号拡張されると、最上位ビット"1"が上位側に複写されるため、H' 00008000ではなく、

H' F F F F 8 0 0 0 に拡張される。したがって、上記の命令 (2) においてアドレスレジスタ A 0 に格納されたデータは、H' 0 0 0 0 8 0 0 0 ではなく、H' F F F F 8 0 0 0 となる。このため、図 1 0 に示すように、次の命令 (3) においては、メモリから読み出されるデータは、H' 0 0 0 8 0 0 0 番地のデータではなく、H' F F F F 8 0 0 0 番地のデータが読み出される。このために、誤ったデータが読み出される。

【0 0 1 4】

また、命令 (6) においても、即値データ H' 8 1 0 0 が符号拡張されると、H' 0 0 0 0 8 1 0 0 ではなく、H' F F F F 8 1 0 0 に拡張される。従って、誤った演算が行われる。

このような即値データの誤拡張は、図 1 0 に示すように、即値データがメモリの全アドレス空間の内、MSB (最上位ビット) の値が "1" のアドレスを指定する場合に発生する。

【0 0 1 5】

このような状態を解消するためには、即値データの指定を 3 2 ビット表記することが考えられる。例えば、命令 (2) においては、即値データを H' 0 0 0 0 8 0 0 0 と指定すればよい。しかしながら、このような指定方法は、1 6 ビットデータを指定する場合でも 3 2 ビット表記する必要が生じるため、命令の表記およびオブジェクトコードが長くなり好ましくない。

【0 0 1 6】

このような問題点を解消するために、1 6 ビット即値データを使用して図 8 に示すプログラムを書き直す方法が図 1 1 に示されている。図 1 1 において、図 8 に示す命令 (2) に該当する命令は、命令 (2-1)、(2-2) の 2 ステップで行われる。

まず、命令 (2-1) で与えられた即値データ H' 8 0 0 0 は、符号拡張されて H' F F F F 8 0 0 0 となる。そして、命令 (2-2) では、拡張データの上位 1 6 ビット分を 0 にクリアするために、拡張データ H' 0 0 0 0 F F F F と H' 0 0 0 0 F F F F との論理積を取り、3 2 ビットデータ H' 0 0 0 0 8 0 0 0 に変換している。

【0017】

また、図8の命令(6)に該当する命令は、図11において命令(6-1)、(6-2)、(6-3)の3ステップで行われる。

命令(6-1)で与えられた即値データH' 8100は、符号拡張されてH' FFFF8100となり、アドレスレジスタA1に格納される。次に命令(6-2)では、拡張データの上位16ビット分を0にクリアするために、拡張データH' FFFF8100とH' 0000FFFFとの論理積を取り32ビットデータH' 00008100に変換する。その後、命令(6-3)において、アドレスレジスタA0とA1とを比較している。

【0018】

しかしながら、この方法では、16ビット即値データの使用が可能なものの、図8に示す方法に比べてプログラムのステップ数が増加してしまうという新たな問題がある。

このように、従来の2つの例は、いずれもアドレスレジスタのデータ幅よりも短い即値データを使用してメモリのアドレス空間にあるデータに対して正確にアクセスするようなプログラム命令を効率良く実行することができないという問題点を有していた。

【0019】

したがって、本発明は上記のような問題点を解消するためになされたもので、全アドレス空間の先頭から2のM乗の範囲内のアドレス空間にあるデータをアクセスする場合に、アドレスレジスタのデータ幅よりも短いデータを即値データとして与えることが可能なデータ処理装置およびデータ処理方法を提供することを目的とする。

【0020】

【課題を解決するための手段】

請求項1の発明に係るデータ処理装置は、整数Nビット幅を有する第1レジスタと、Nビット幅を有する第2レジスタと、Nより小さいM(整数)ビットデータの最上位ビットを上位側へ複写することによってMビットデータのデータ幅をNビットに拡張する符号拡張器と、Mビットデータの上位側にゼロ値を発生させ

ることによってMビットデータのデータ幅をNビットに拡張するゼロ拡張器と、与えられた命令に従って演算処理を行う演算器と、命令を解読し、命令がMビットの即値データを含み、かつ第1レジスタに対する格納処理を有する場合、即値データを前記ゼロ拡張器に出力し、命令がMビットの即値データを含み、かつ第2レジスタに対する格納処理を有する場合、即値データを符号拡張器に出力するように第1レジスタ、第2レジスタ及び演算器を制御する命令制御手段とを備える。

【0021】

請求項2の発明に係るデータ処理装置は、請求項1の発明に対し、第1レジスタが、メモリのアドレスデータを格納するアドレスレジスタであり、第2レジスタが、命令のデータオペランドを格納するデータレジスタである。

請求項3の発明に係るデータ処理装置は、請求項1の発明に対し、命令制御手段が、2のべき乗の即値データを解読する。

【0022】

請求項4の発明に係るデータ処理方法は、整数Nビット幅の第1及び第2レジスタに対してアクセスするNより小さい整数Mビット幅の即値データを指定した命令に従って所定の演算を行う方法であって、命令を解読し、命令が指定するアクセス先レジスタを判別する命令解読ステップと、命令が第1レジスタに対する格納処理を有する場合、Mビット幅の即値データをゼロ拡張することによってNビット幅のデータに拡張し、命令が第2レジスタに対する格納処理を有する場合、Mビット幅の即値データを符号拡張することによってNビット幅のデータに拡張するデータ拡張ステップと、Nビット幅に拡張された即値データを用い、命令に従って所定の演算処理を行う演算ステップとを備える。

【0023】

請求項5の発明に係るデータ処理方法は、請求項4の発明に対し、第1レジスタが、メモリのアドレスデータを格納するアドレスレジスタであり、第2レジスタが、命令のデータオペランドを格納するデータレジスタである。

請求項6の発明に係るデータ処理方法は、請求項4の発明に対し、即値データのビット幅Mが、2のべき乗で指定される。

【0024】

【作用】

請求項1乃至請求項6の発明において、命令制御手段は、即値データを含む命令を解読する。そして、即値データが第1レジスタに対してアクセスされるデータである場合、即値データをゼロ拡張器に出力する。また、即値データが第2レジスタに対してアクセスされるデータである場合には、即値データを符号拡張器に出力する。ゼロ拡張器及び符号拡張器は、与えられた即値データに対して、各々ゼロ拡張及び符号拡張処理を行い、Mビット幅の即値データをNビット幅のデータに拡張する。その後、演算器は、拡張された即値データを使用し、所定の演算処理を行う。

【0025】

【実施例】

以下、本発明の実施例について、図面を参照しながら説明する。

図1は、本発明の実施例によるデータ処理装置の全体構成を示すブロック図であり、図2はその詳細ブロック図である。図1及び図2において、データ処理装置は、外部との命令及びデータ転送を制御するバス転送制御部101と、命令読み出し動作を制御する命令フェッチ部102と、命令を解読する命令解読部103と、32ビット幅のデータ演算を行う演算器104と、演算器104が使用するデータを格納するレジスタ105と、演算器104での演算結果に基づいて設定されるフラグ群を格納するプログラム状態語106と、分岐命令の場合に条件成立か不成立かを判断する分岐判断部107と、実行中の命令アドレスを格納するプログラムカウンタ108と、16ビット入力データの最上位ビットを符号拡張して32ビットデータとして出力する符号拡張器109と、16ビット入力データの上位側16ビット分に値ゼロを発生させて32ビットに拡張するゼロ拡張器110とを備えている。

【0026】

レジスタ105は、主としてデータを格納する32ビット幅の8個のレジスタD0～D7からなるデータレジスタ群105aと、主としてアドレスデータを格納する32ビット幅の7個のレジスタA0～A6からなるアドレスレジスタ群1

05bとを備えている。

以上のように構成されたデータ処理装置について、その動作を説明する。なお、図3は、主に命令解読部103の処理フローを示している。

【0027】

まず、命令フェッチ部102は、バス転送制御部101を起動し、外部のROM等に格納されたプログラムの命令を逐次読み出す。読み出されたプログラムの命令は、命令フェッチ部102を介して命令解読部103に出力される。

図3を参照して、命令解読部103は、読み出された命令を解読する（ステップS10）。命令の解読動作において、算術演算、論理演算、レジスタ間転送、メモリ、分岐等の命令の種類を示すオペレーションコード及びデータ、即値データ、レジスタ番号、メモリアドレス等の演算数（オペランド）が解読される。この解読結果に従って、この命令が32ビットより短いビット幅の即値データを指定しているか否かを判断する（ステップS20）。短いビット幅の即値データを指定していれば、次のステップS30に進み、指定していなければステップS40に進む。

【0028】

短いデータ幅の即値データが検出されると、この命令が指定するアクセス先レジスタ（ディスティネーションレジスタ）がデータレジスタ群105aにあるかアドレスレジスタ群105bにあるかを判別する（ステップS30）。

さらに、制御信号発生部103aは、解読された命令の内容及びアクセス先のレジスタの種別に応じて演算器104、符号拡張器109、ゼロ拡張器110に制御信号を出力する（ステップS40）。

【0029】

ここで、命令の内容毎にデータ処理装置の処理動作を説明する。

(1) 「データレジスタ群105aまたはアドレスレジスタ群105bに属するレジスタ間のデータ転送、あるいはこれらのレジスタに格納されたデータの演算」命令の場合。（例えば、“MOVE D0,D1”や“ADD A0,A1”等）

前者では、ソースレジスタから読み出されたNビット幅のデータが指定された

レジスタ（ディスティネーションレジスタ）に格納される。

【0030】

後者では、所定のレジスタから読み出された2つのNビット幅のデータが演算器104に入力され、所定の演算がなされた後、指定されたレジスタに格納される。

(2) 「Mビットの即値データをデータレジスタ群105aに格納し、あるいは演算する」命令の場合。（例えば、“MOVI #H' 0010, D0”や“ADDI #H' 0010, D0”等）

前者では、命令解読部103において、命令がMビット幅の即値データを指定し、かつディスティネーションレジスタがデータレジスタ群105aであることが解読される。この結果、Mビット幅の即値データは、符号拡張器109でNビット幅に拡張されて指定されたデータレジスタへ格納される。

【0031】

後者では、上記と同様に、Mビット幅の即値データは、符号拡張器109でNビット幅に拡張されて演算器104の一方に入力される。そして、所定の演算が行われた後、指定されたデータレジスタへ出力される。

(3) 「Mビットの即値データをアドレスレジスタ群105bに格納し、あるいは演算する」命令の場合。（例えば、“MOVI #H' 0010, A0”や“ADDI #H' 0010, A0”等）

前者では、命令解読部103において、命令がMビットの即値データを指定し、かつディスティネーションレジスタがアドレスレジスタ群105bであることが解読される。この結果、Mビット幅の即値データは、ゼロ拡張器110でNビット幅に拡張されて指定されたアドレスレジスタへ出力される。

【0032】

後者では、同様にMビット幅の即値データがゼロ拡張器110でNビット幅にゼロ拡張されて演算器104の一方に入力される。そして、所定の演算が行われた後、指定されたアドレスレジスタ群105bへ出力される。

ここで、ゼロ拡張器110の動作について図4を用いて説明する。ゼロ拡張器110は、Mビットで表現される数値の最上位ビットが0、1のいずれであって

も、上位側へNビットまで0を発生させて拡張する。

【0033】

さらに、この実施例によるデータ処理装置について、特に、Mビット幅の即値データの処理動作を具体例を用いて説明する。一例として、従来の装置の説明に使用したものと同一内容のプログラムの実行動作について説明する。

図6は、従来の例を示す図8及び図11に対応するプログラムを示している。このプログラムの内容を再記すると、図10に示すように、32ビットでアドレス指定されるメモリの全アドレス空間の中でH' 8000 (H' は16進数表記を示す) 番地からH' 8100番地までH' 10番地刻みに格納されているメモリ上の16個のデータを順次加算し、その和をH' 10000000番地へ書き出す処理を示している。以下、図6に従って図1～図5を参照して説明する。

命令(1) : データレジスタD0をクリアする。

命令(2) : アドレスレジスタA0に32ビットの即値データH' 00008000を設定する。

【0034】

図3を参照して、命令解読部103は、この命令を解読する(ステップS10)。そして、16ビット表記の即値データH' 8000を使用していることを判別する(ステップS20)。さらに、アドレスレジスタ群105bへのデータ格納命令であることを判別する(ステップS30)。したがって、命令解読部103は、制御信号発生部103aによって、ゼロ拡張器110に制御信号を発する(ステップS40)。ゼロ拡張器110は、制御信号に従って16ビットの即値データH' 8000を32ビットデータH' 00008000にゼロ拡張し、アドレスレジスタA0にロードする。

命令(3) : アドレスレジスタA0で指定されるメモリの内容をデータレジスタD1に格納する。

命令(4) : データレジスタD1に格納されたデータをデータレジスタD0に格納されたデータに加算し、データレジスタD0に格納する。

命令(5) : アドレスレジスタA0に格納されたデータに即値データH' 0010を加算し、アドレスレジスタA0に格納する。

【0035】

この命令は16ビットの即値データH' 0010を使用している。そして、アドレスレジスタA0へのデータ格納を指示している。したがって、命令解読部103は、制御信号発生部103aによって、先ずゼロ拡張器110に制御信号を発する。ゼロ拡張器110は、制御信号に従って16ビットの即値データH' 0010を32ビットデータH' 00000010にゼロ拡張し、演算器104の一方に出力する。

【0036】

また、制御信号発生部103aからの制御信号を受けて、アドレスレジスタA0は、格納されたデータH' 00008000を読み出し、演算器104の他方に出力する。

さらに、演算器104は、入力された2つの32ビットデータ(H' 00008000、H' 00000010)を加算し、加算結果をアドレスレジスタA0に格納する。

命令(6)：上記の算出データと即値データH' 8100とを比較する。

【0037】

この命令は、16ビットの即値データH' 8100を使用している。そして、アドレスレジスタA0のデータとの比較を指示している。したがって、制御信号発生部103aは、先ずゼロ拡張器110に制御信号を発する。ゼロ拡張器110は、制御信号に従って16ビットの即値データH' 8100を32ビットデータH' 00008100にゼロ拡張し、演算器104の一方に出力する。

【0038】

また、アドレスレジスタ105bは、制御信号発生部103aからの制御信号を受けて、アドレスレジスタA0から格納データ(H' 00008010)を読み出し、演算器104の他方側に出力する。

演算器104は、入力された2つの32ビットデータを比較する。

命令(7)：比較結果に基づいて、小さければラベルAで示す命令(3)へ戻り、等しいかまたは大きければ命令(8)へ進むように判断する。

【0039】

命令（3）から命令（7）で記述されるループ処理は、アドレスレジスタA0の値がH' 00008000から始めてH' 00000010ずつ更新され、H' 00008100になるまで繰返される。このために、16回の加算結果がデータレジスタD0に格納された後、命令（8）へ進む。

命令（8）：データレジスタD0の内容をメモリのH' 10000000番地へ格納する。

【0040】

このように、命令（2）及び命令（6）において使用した即値データは、図10に示すメモリの先頭から2の16乗のアドレス空間のアドレスデータを16ビット表記で指定することができる。このために、32ビット表記する必要がある従来例に比べて、命令（2）、（6）の2ステップにおいて即値データの記述がそれぞれ2バイト（16ビット）分短くなる。その結果、上記例のプログラムの場合には、従来例に比較してプログラムサイズを4バイト小さくすることができる。

【0041】

さらに、同一バスで命令とデータがアクセスされる構成のデータ処理装置においては、プログラムサイズの削減は、命令獲得のためのバスアクセスと命令実行に伴うデータアクセスのためのバスアクセスとの競合を減少させることになり、実行速度の向上に寄与することができる。

なお、上記実施例においては、32ビットのアドレスレジスタを有するデータ処理装置に対して、16ビットの即値データの処理について説明したが、この例に限定されるものではなく、16、32等Nビットデータの格納が可能なアドレスレジスタに対して、4、8、16等M（ $M < N$ ）ビットの即値データの処理に適用することができる。

【0042】

【発明の効果】

このように、本発明は、即値データを含む命令が与えられた場合に、即値データがアクセス対象とするレジスタの種別に応じて、ゼロ拡張及び符号拡張のいず

れかを判別してデータ拡張した後、所定の演算処理を行うように構成されているので、Nビットで表現される全アドレス空間のMSBが”1”のアドレス空間にあるデータをアクセスするための即値データとして、Nより小さいMビットの即値データを指定することができる。これにより、即値データの表記が簡略化され、プログラムサイズを小さくすることができる。

【図面の簡単な説明】

【図1】

本発明の実施例によるデータ処理装置の全体構成を示すブロック図である。

【図2】

図1に示すデータ処理装置の主要部の構成を示すブロック図である。

【図3】

本実施例のデータ処理装置の命令解読部103の処理を示す処理フロー図である。

【図4】

ゼロ拡張器の動作を概念的に示す説明図である。

【図5】

符号拡張器の動作を概念的に示す説明図である。

【図6】

本実施例のデータ処理装置に使用されるプログラム例を示す説明図である。

【図7】

従来の第1例におけるデータ処理装置の構成を示すブロック図である。

【図8】

図7に示す従来のデータ処理装置に使用されるプログラム例を示す説明図である。

【図9】

図8に示すプログラムの動作を説明するフローチャートである。

【図10】

データ処理装置が使用するメモリ空間を模式的に示す説明図である。

【図11】

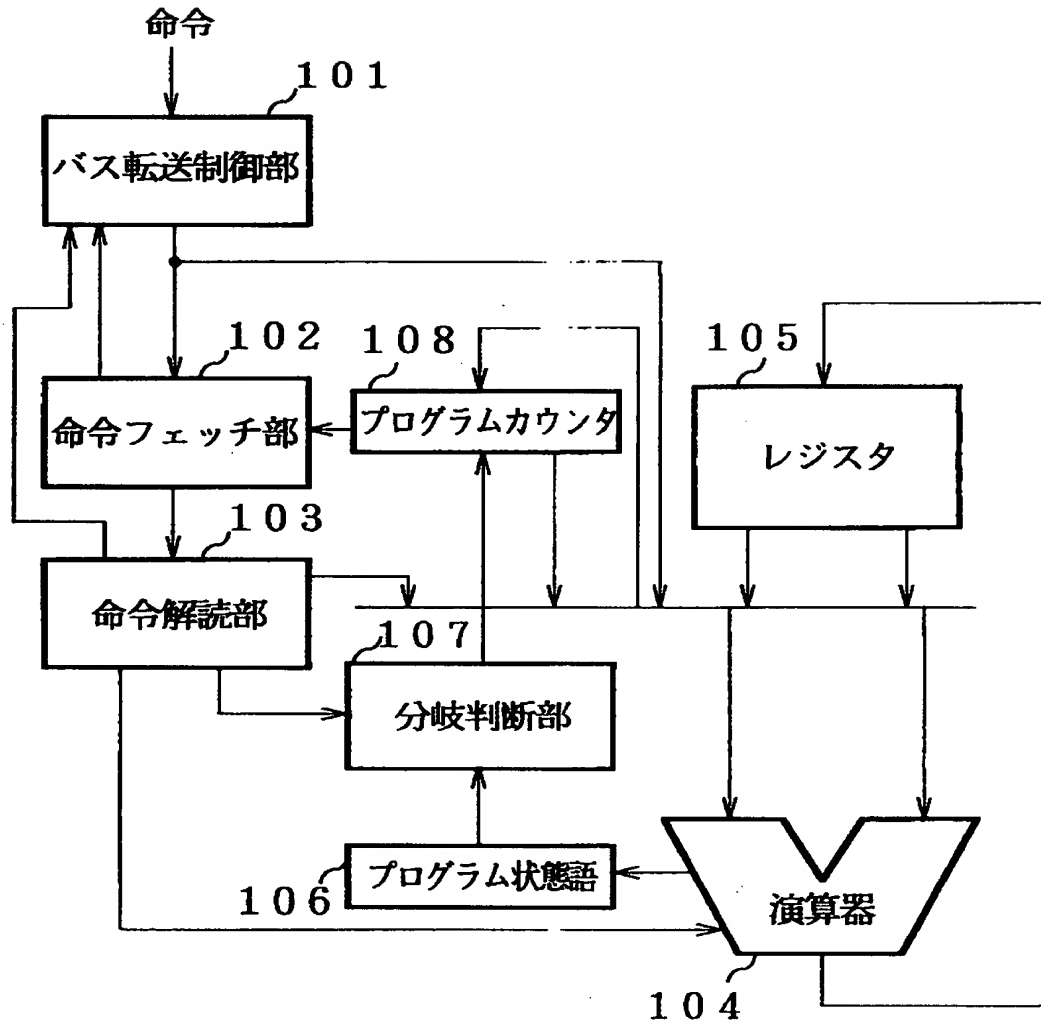
従来の第2例によるプログラム例を示す説明図である。

【符号の説明】

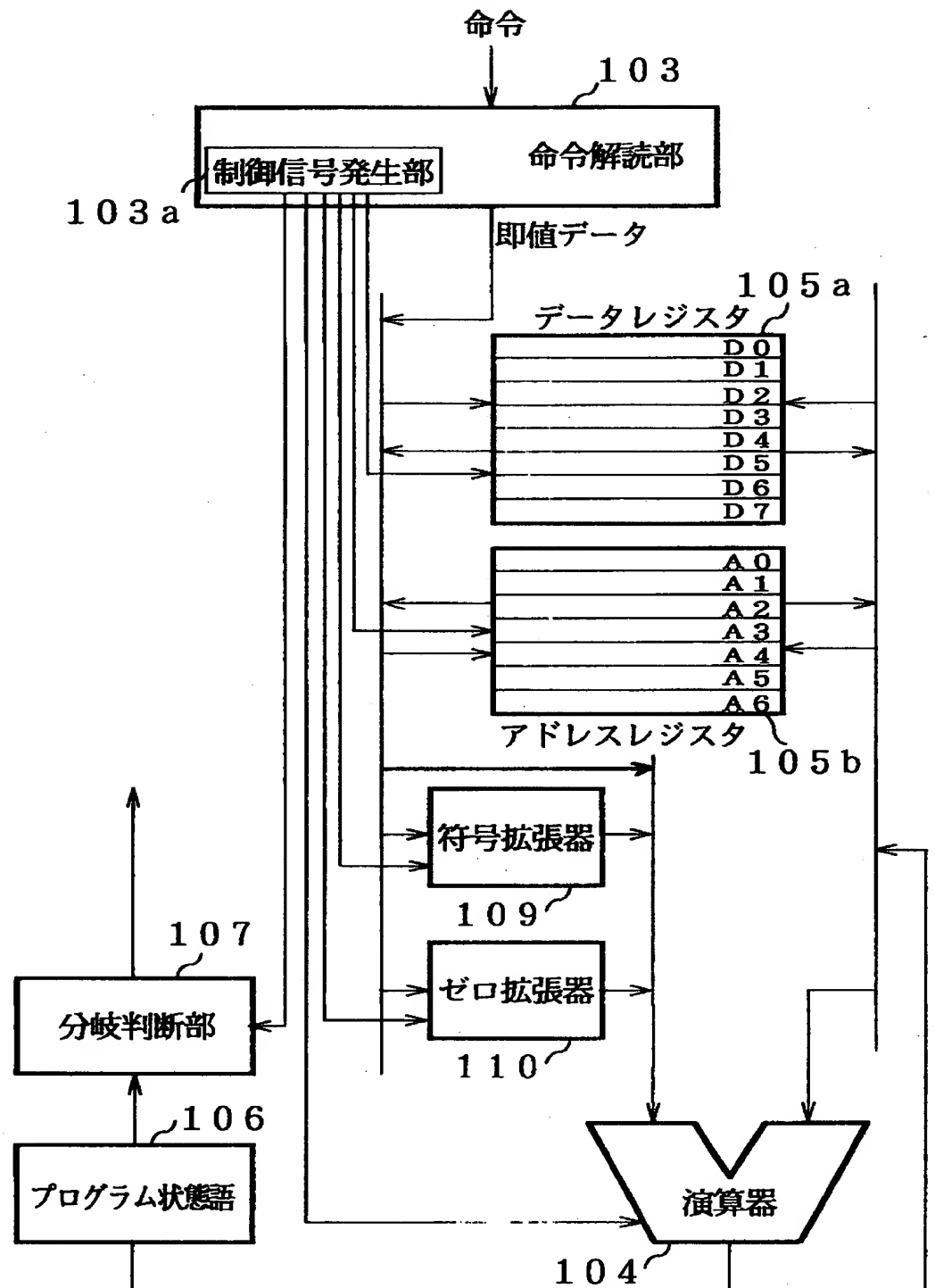
- 103 命令解読部
- 103a 制御信号発生部
- 104 演算部
- 105 レジスタ
- 105a データレジスタ群
- 105b アドレスレジスタ群
- 109 符号拡張器
- 110 ゼロ拡張器

【書類名】 図面

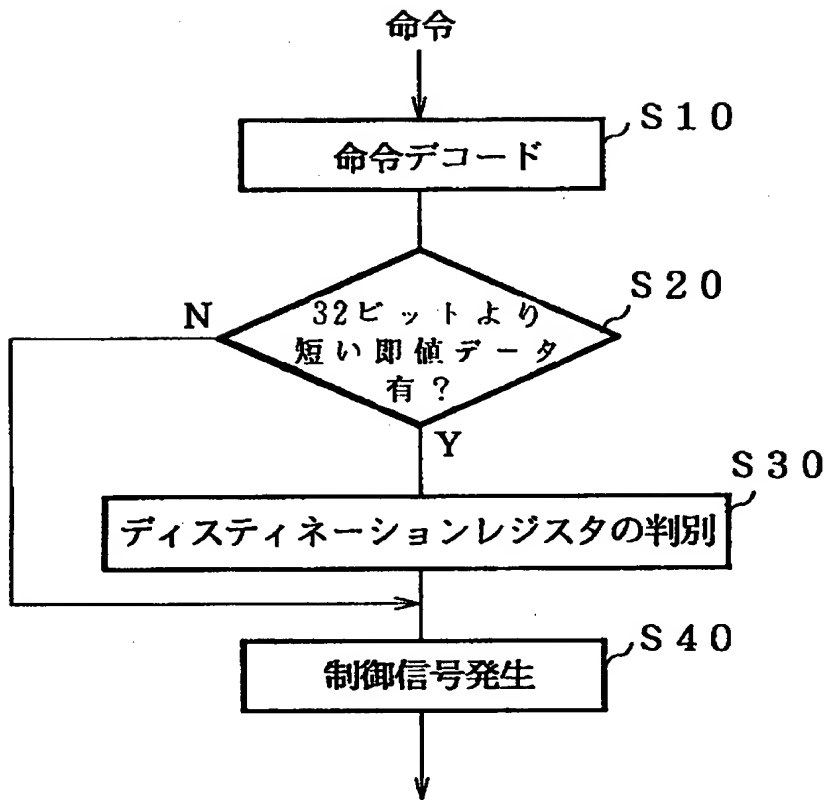
【図1】



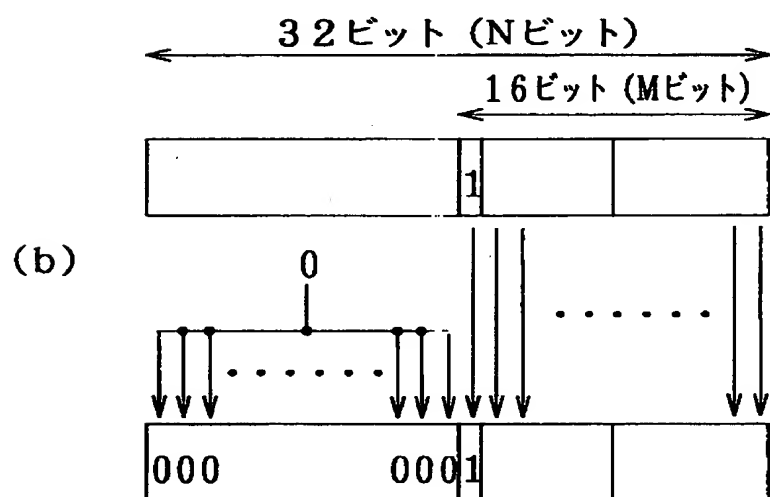
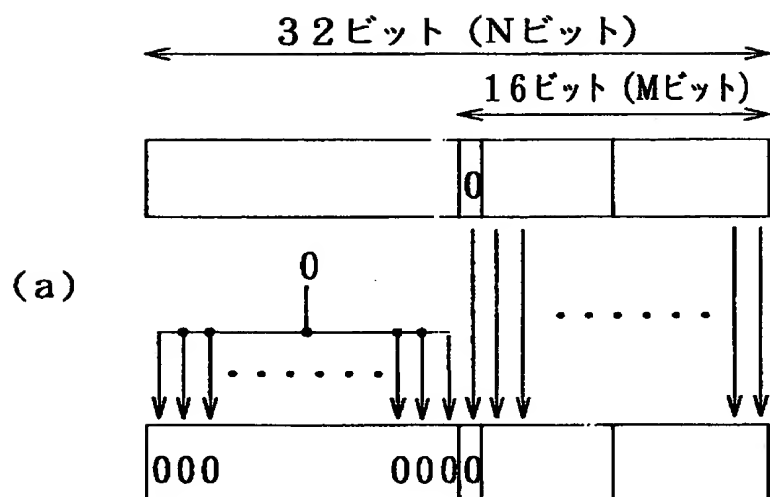
【図2】



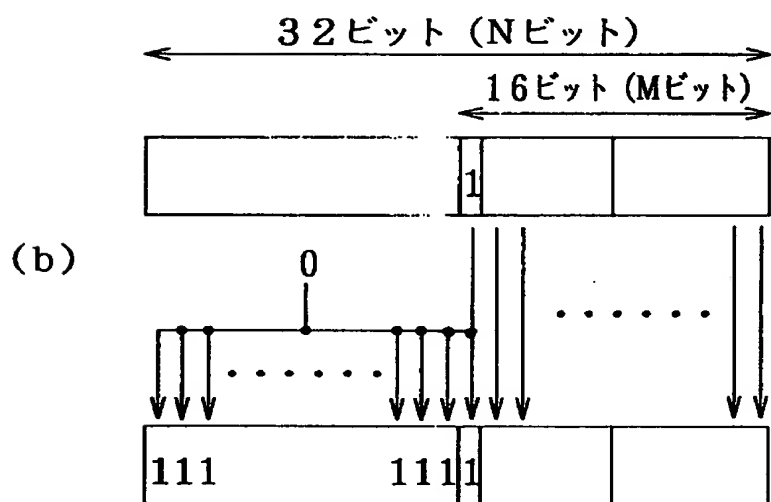
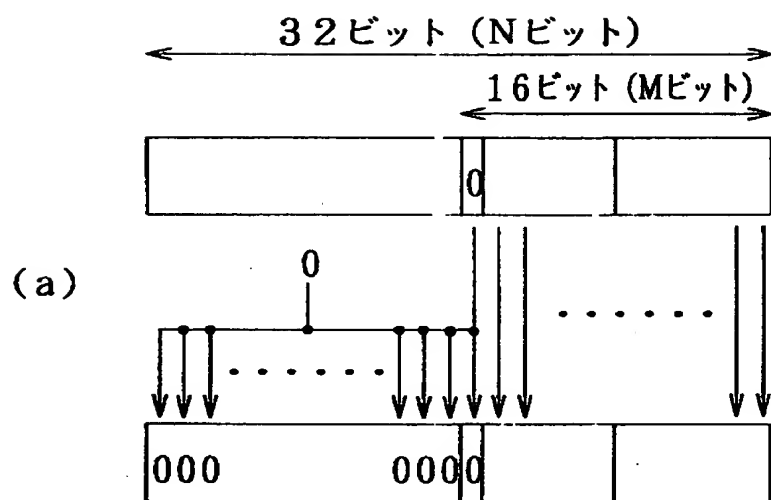
【図3】



【図4】



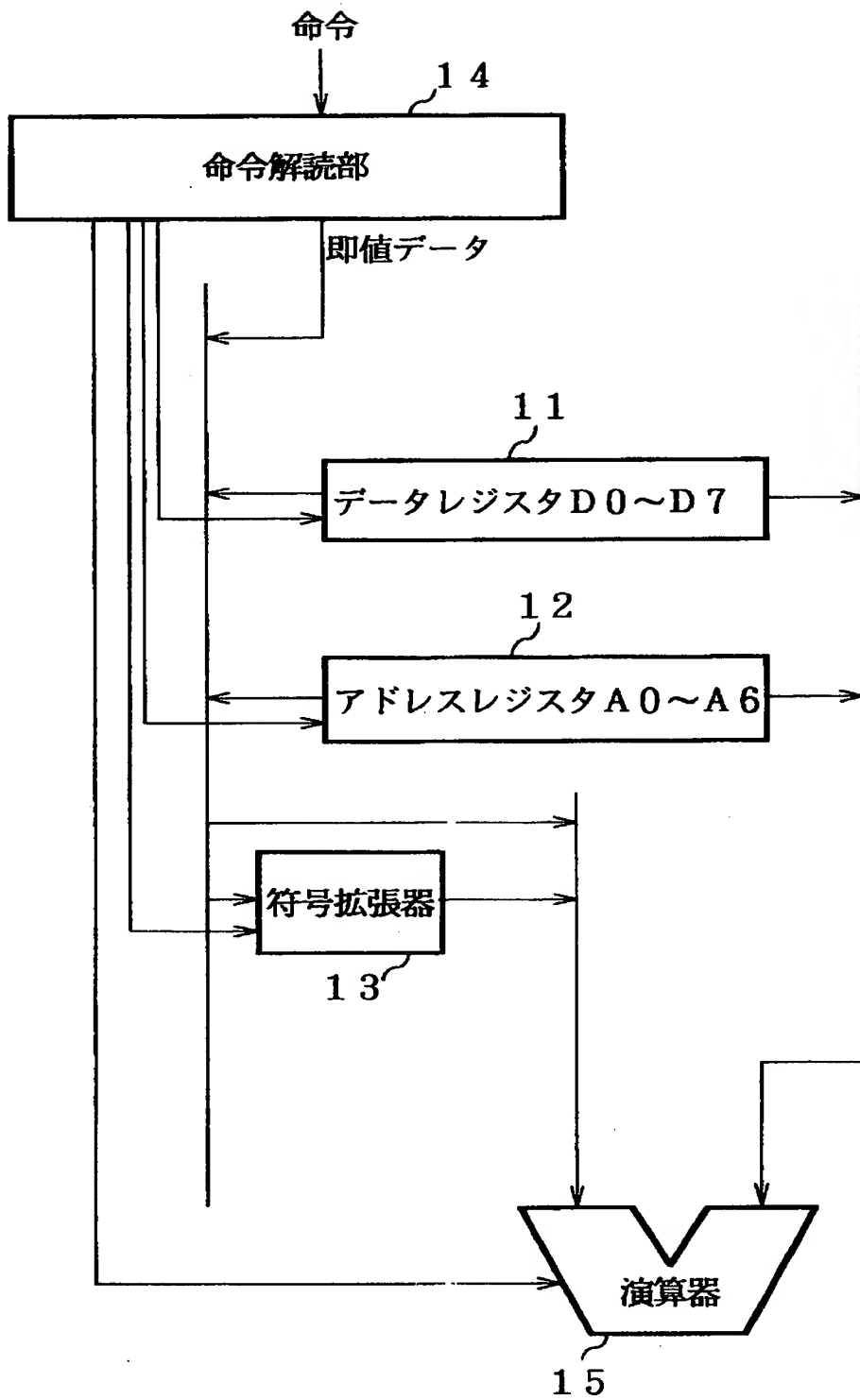
【図5】



【図6】

SUB D0, D0	—	(1)
MOVI #H' 8000, A0	—	(2)
ラベルA		
MOV @A0, D1	—	(3)
ADD D1, D0	—	(4)
ADDI #H' 0010, A0	—	(5)
CMPI #H' 8100, A0	—	(6)
BLT ラベルA	—	(7)
ラベルB		
MOV D0, @H' 10000000	—	(8)

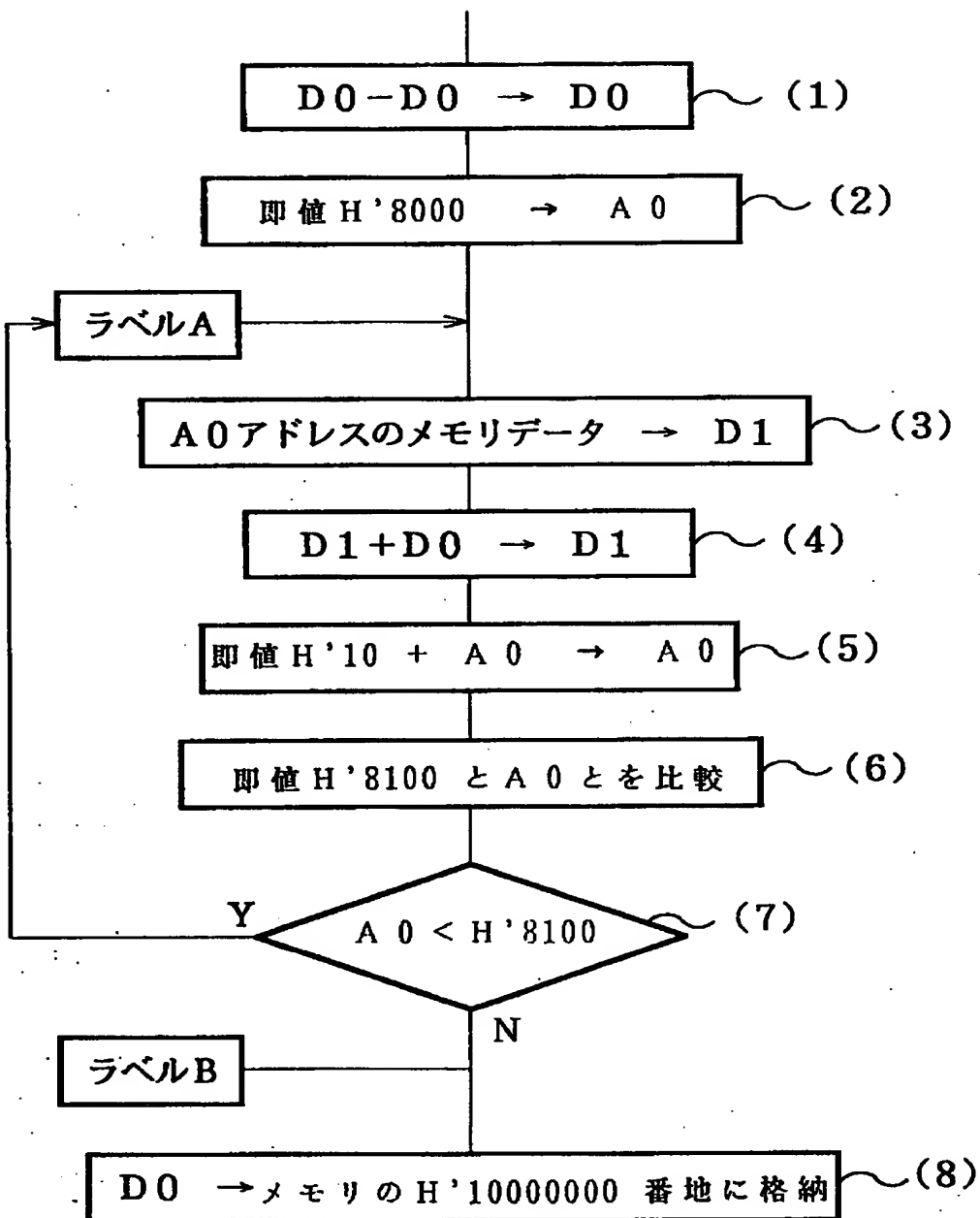
【図 7】



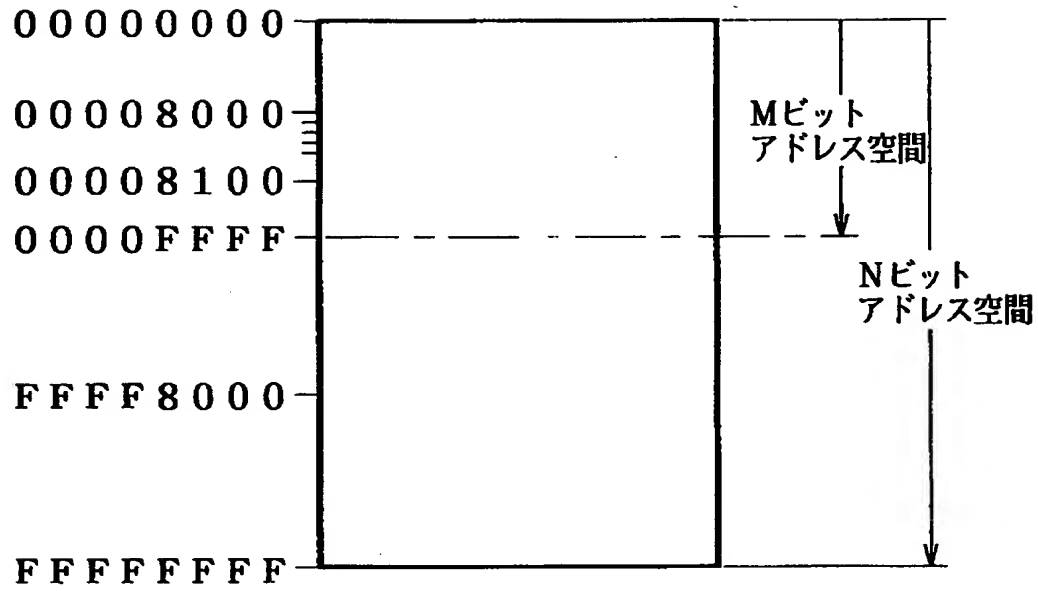
【図 8】

SUB D0, D0	—	(1)
MOVI #H' 8000, A0	—	(2) → MOVI #H' 00008000, A0
ラベルA		
MOV @A0, D1	—	(3)
ADD D1, D0	—	(4)
ADDI #H' 0010, A0	—	(5) → ADDI #H' 00000010, A0
CMPI #8100, A0	—	(6) → CMPI #H' 00008100, A0
BLT ラベルA	—	(7)
ラベルB		
MOV D0, @H' 10000000	—	(8)

【図9】



【図10】



【図11】

SUB D0, D0	—	(1)
MOVI #H' 8000, A0	—	(2-1)
ANDI #H' 0000FFFF, A0	—	(2-2)
ラベルA		
MOV @A0, D1	—	(3)
ADD D1, D0	—	(4)
ADDI #H' 0010, A0	—	(5)
MOVI #H' 8100, A1	—	(6-1)
ANDI #H' 0000FFFF, A1	—	(6-2)
CMP A1, A0	—	(6-3)
BLT ラベルA	—	(7)
ラベルB		
MOV D0, @H' 10000000	—	(8)

【書類名】 要約書

【要約】

【目的】 データ処理装置において、メモリのMSBが”1”のアドレス空間にあるデータをアクセスする場合に、Mビットの即値データの使用によるプログラムの簡略化を可能とする。

【構成】 データ処理装置は、Nビット幅のデータレジスタ群105aと、Nビット幅のアドレスレジスタ群105bと、M ($M < N$) ビットデータをNビットデータに拡張する符号拡張器109と、MビットデータをNビットデータに拡張するゼロ拡張器110と、命令に従って演算処理を行う演算器104と、命令を解読する命令解読部103と、命令がMビットの即値データを含み、かつアドレスレジスタ群105bへのアクセス対象データである場合、即値データをゼロ拡張器110に出力し、データレジスタ群105aへのアクセス対象データである場合、即値データを符号拡張器109に出力する制御信号を発信する制御信号発生部103aとを備える。

【選択図】 図2

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000005821

【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地

【氏名又は名称】 松下電器産業株式会社

【代理人】 申請人

【識別番号】 100090446

【住所又は居所】 大阪府大阪市北区豊崎 3 丁目 2 番 1 号 淀川 5 番館

5 F 中島国際特許事務所

【氏名又は名称】 中島 司朗

出 願 人 履 歴 情 報

識別番号 [000005821]

1. 変更年月日	1990年 8月28日
[変更理由]	新規登録
住 所	大阪府門真市大字門真1006番地
氏 名	松下電器産業株式会社